I. System Functionality

The text categorization program was developed using Anaconda's Spyder, with programming language Python 3.11. As mentioned in **naiveBayesClassifier.py,** libraries needed for this program include `OS` (standard library), `String` (standard library), `numpy` (`pip install numpy`), `collections` (standard library), and `NLTK`, along with `NLTK` packages "`punkt`" and "`stopwords`":

```
# nltk.download('punkt')

# nltk.download('stopwords')
```

In order to run the program:

1. Download the given **TC_provided** folder into an available directory.
2. Run **naiveBayesClassifier.py**.
3. 1st input example: "Enter the filename that contains labeled training documents: corpus1_train.labels."
4. 2nd input example: "Enter the filename that contains unlabeled test documents: corpus1_test.list."
5. Run **analyze.pl** to compare predicted model's accuracy: "`!perl analyze.pl predicted_corpus1_test.labels corpus1_test.labels`"

II. ML Method: Naive Bayes

Naive Bayes was utilized to implement text categorization. The system tokenized training and test files through the *tokenizeDoc* function, which included converting document's content to lowercase, and removing punctuation. This reduced the vocabulary size, and improved the generalization of the system. Next, the function applies tokenization using *`nltk.word_tokenize`*. Afterwards, the algorithm filtered stopwords, or common words such as "the" and "is," using the `nltk.corpus.stopwords list`. This greatly improved the model's accuracy by excluding non-informative words, and hence generating word probabilities better. Lastly, the function incorporated stemming using NLTK's `PorterStemmer`. This significantly improved the model's accuracy due to the improvement in generalizing different word forms. Specifically, its inclusion was useful for corpus 2&3, which were larger datasets.

The system implements Laplace smoothing in the *calcLikelihood* function, with alpha = 0.056, which was determined from multiple tests. The value of alpha is small to prevent excessive smoothing, which tends to occur when alpha = 1.

Lastly, the system's performance for corpus 2 & 3 are evaluated with the **creatingTestSet_SubTraining.py** algorithm, which utilizes split ratios to divide the provided training sets into "sub-training sets," and a "test set." The user was prompted to enter a split ratio between 0 and 1. For the *Results*, approximately ⅓ of the training test was utilized for testing, while ⅔ was used as the sub-training set.

III. Results

- *Corpus 1*

```
Enter the filename that contains labeled training documents: corpus1_train.labels
Enter the filename that contains unlabeled test documents: corpus1_test.list

In [65]: !perl analyze.pl predicted_corpus1_test.labels corpus1_test.labels
Processing answer file...
Found 5 categories: Pol Str Oth Dis Cri
Processing prediction file...

394 CORRECT, 49 INCORRECT, RATIO = 0.889390519187359.

CONTINGENCY TABLE:
        Pol     Str     Oth     Dis     Cri     PREC
Pol     123     4       5       0       1       0.92
Str     18      128     3       1       7       0.82
Oth     2       0       13      0       0       0.87
Dis     0       1       3       88      0       0.96
Cri     1       2       1       0       42      0.91
RECALL  0.85    0.95    0.52    0.99    0.84

F_1(Pol) = 0.888086642599278
F_1(Str) = 0.876712328767123
F_1(Oth) = 0.65
F_1(Dis) = 0.972375690607735
F_1(Cri) = 0.875
```

- *Corpus 2*

```
Enter the filename that contains labeled training documents: corpus2_trainingsub.labels
Enter the filename that contains unlabeled test documents: corpus2_testing.list

In [68]: !perl analyze.pl predicted_corpus2_test.labels corpus2_testing.labels
Processing answer file...
Found 2 categories: I O
Processing prediction file...

246 CORRECT, 50 INCORRECT, RATIO = 0.831081081081081.

CONTINGENCY TABLE:
        I       O       PREC
I       60      15      0.80
O       35      186     0.84
RECALL  0.63    0.93

F_1(I) = 0.705882352941177
F_1(O) = 0.881516587677725
```

- *Corpus 3*

```
Enter the filename that contains labeled training documents: corpus3_subtain.labels
Enter the filename that contains unlabeled test documents: corpus3_test.list

In [70]: !perl analyze.pl predicted_corpus3_test.labels corpus3_test.labels
Processing answer file...
Found 6 categories: Fin Sci USN Spo Wor Ent
Processing prediction file...

219 CORRECT, 20 INCORRECT, RATIO = 0.916317991631799.

CONTINGENCY TABLE:
        Fin     Sci     USN     Spo     Wor     Ent     PREC
Fin     25      0       1       0       0       0       0.96
Sci     1       28      0       0       1       2       0.88
USN     2       0       57      1       5       0       0.88
Spo     0       0       0       28      0       1       0.97
Wor     0       0       2       0       74      1       0.96
Ent     1       0       0       1       1       7       0.70
RECALL  0.86    1.00    0.95    0.93    0.91    0.64

F_1(Fin) = 0.909090909090909
F_1(Sci) = 0.933333333333333
F_1(USN) = 0.912
F_1(Spo) = 0.949152542372881
F_1(Wor) = 0.936708860759494
F_1(Ent) = 0.666666666666667
```